CIRP General Assembly 2019

Contents lists available at SciVerse ScienceDirect

CIRP Annals Manufacturing Technology

Journal homepage: www.elsevier.com/locate/cirp

Degradation-aware decision making in reconfigurable manufacturing systems

Xingyu Li[1], Aydin Nassehi[2], Bogdan I. Epureanu[1]*

1. Department of Mechanical Engineering, University of Michigan, Ann Arbor, USA
2. Department of Mechanical Engineering, University of Bristol, UK
** Corresponding author*

Reconfigurable manufacturing systems (RMS) are designed to improve responsiveness and adaptability to individualized demands, creating a potential solution for mass personalization. System reconfigurations provide flexibility to fluctuating demands, and can be enhanced by adjustments of machine components. However, improper balancing between maintenance and reconfiguration actions can result in system breakdowns and can hamper system health and ability to reconfigure. This paper proposes a degradation-aware RMS decision-making model to optimally determine and adjust operational actions in real-time considering demand fulfilment, maintenance cost, and system health. The proposed approach has the capability to capture the causality between operational action sequences and the resulting system deterioration through artificial intelligence-based methods.

Decision making, Reconfiguration, Manufacturing systems

1. Introduction

This paper presents a novel method to take into account module degradation in the configuration optimization and reconfiguration of manufacturing systems. Reconfigurable manufacturing systems (RMSs) [1] combine the benefits of flexible manufacturing systems and flowshops. These systems comprise machines that can be reconfigured to modify their functionality. Such machines are made from discrete modules. By changing the modules or by changing the module configuration in machines, RMSs can be reconfigured to achieve a different functionality. In RMS operation, there is a need to determine a suitable mechanism for optimizing the configuration [2] and triggering reconfigurations [3]. Currently, there is increasing pressure on manufacturing systems to cope with variable customer demand at minimum cost requiring more flexibility [4]. This combined with new paradigms such as cloud manufacturing where reconfiguration is observed at a large scale, makes research on RMS more relevant today than it has ever been [5]. Traditionally, in optimal configuration of RMSs, modules are assumed to be healthy; module degradation is not considered. Maintenance planning [6] and particularly predictive maintenance [7], however, are important determining factors for the total cost of a manufacturing system. Recent research on maintenance planning has indicated the importance of taking into account the condition of the elements of the system [8] to predict the future states of the system [9]. Multivariate analysis techniques [10] and cloud enabled prognosis [11] are amongst more recent developments in this field. Data driven approaches such as those based on artificial intelligence (AI) have shown excellent potential for this purpose [12]. In this paper, the work on maintainability of RMSs [13] is recognized, and an extended framework for reconfiguration of RMSs taking into account module degradation is provided. AI methods are used with the goal of achieving robust RMS operation, resilient to failures due to module degradation.

The main advantage of RMSs is that the flexibility in the machine structure creates opportunities in offering demand-dependent capacity. However, this potential for adaptive capacity comes at the cost of potential added degradation and down time. The additional time required for reconfiguration is heavily dependent on complicated optimization methods but has a major effect on the overall production rate that can be achieved. In addition, reconfiguration actions hasten the degradation of machine modules and shorten their lifetime.

There is, thus, a challenge to balance two efforts: one effort on reconfiguration with the aim of maximizing production rate and demand satisfaction, and the other effort on maintenance to maximize machine module lifetime. Degradation for each module can be defined as the failure probability of that module at any given time. With this definition, module degradation can be taken into account in the reconfiguration of RMSs to achieve this balance. It is noteworthy that the correlation between operational actions and their harm to the module condition is obscure and time-variant. It is thus necessary to capture the relationship in real-time for monitoring the inventory status and making decisions accordingly. In the following sections, an integrated model is introduced to take module degradation into account in the decision making related to RMSs. This will be followed by a case study where the model is applied to an industry-inspired problem. The results are then presented followed by conclusions and aspirations for future work.

2. Model description

An integrated decision-making model is designed to satisfy a stochastic arrival of demands while managing the heath of the RMS. The structure of the model and the associated information flow is shown in Figure 1. At each time t, the system determines if reconfiguration (and hence configuration re-mining, A) is needed based on the received demands and the machine status. If not, then assembly, disassembly, relocation and maintenance (ADLM) actions are scheduled for predictive maintenance and repair if damaged (B). If yes, then reconfiguration is triggered, which starts with the generation of configuration candidates (C). Each configuration is evaluated in two steps: 1) the reconfiguration planning (D) evaluates the best condition that the configuration candidate can achieve given the limited working capacity; 2) the evolution of capacity distribution (E) estimates the probability of different capacity that the configuration candidate may perform in the short future. Given the capacity distributions, the configuration planning (F) selects the candidate with the highest likelihood of demand fulfilment as the new configuration. The determined operational actions are imported to the operation simulator (G) to update action history and module usage history, which are the inputs to a deep neural network for model training and updating the module failure probability (H).

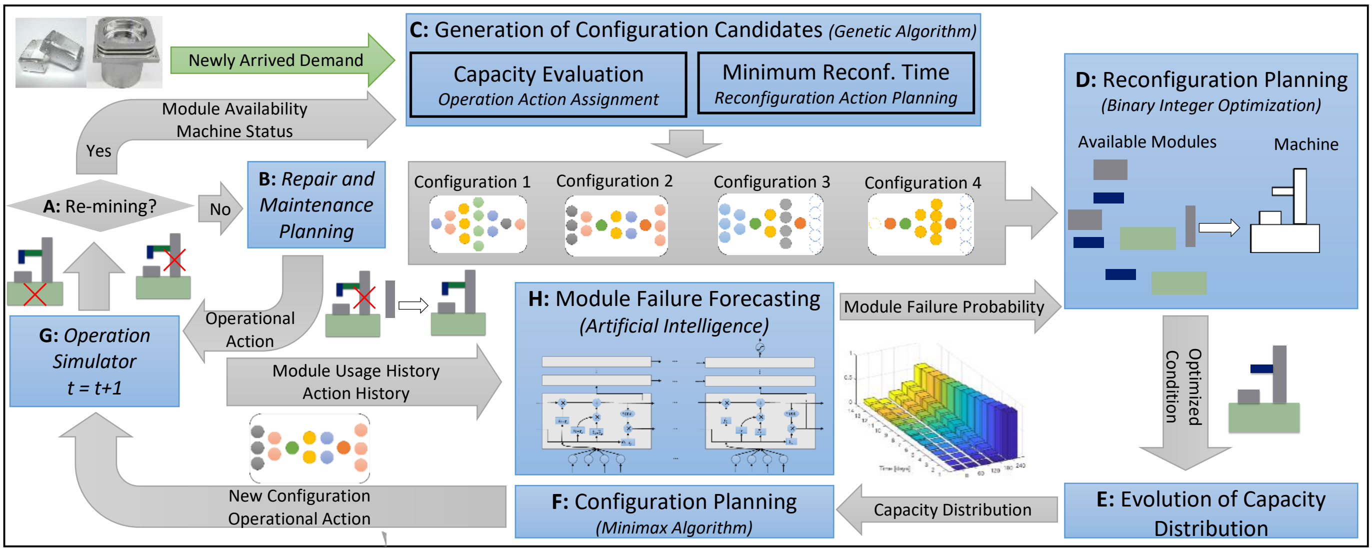

Figure 1. Artificial intelligence-based decision-making model for optimizing the robustness of RMS reconfiguration

2.1. Reconfiguration, Repair and Maintenance Planning

The state of module i is represented by a binary vector $\boldsymbol{s}^i = [s_0^i, s_1^i, s_2^i, s_3^i, \dots, s_{N_s}^i]$, where N_s is the number of machines. s_k^i=1 indicates that module i is in machine k. Modules that are not in machines are in the inventory, denoted by machine $k = 0$. Module i can be removed from machine k and assembled in machine k', (operation action $o_{kk'}^i$). Machine-wise actions are also created to facilitate operations. Machine k, it can be relocated to the location of machine k' (operation action $o_{k'}^k$) or it can be disassembled to inventory (operation action o_0^k). To increase the fidelity of the model, regular and quick actions (denoted by $\boldsymbol{q}$) are considered. Quick actions require less time to accomplish but degrade the module more. Module failure probability can be reduced by p_d^i after disassembly or by p_a^i by maintenance before assembly. Denote a new configuration by $\boldsymbol{S} = [\boldsymbol{s}^1, \boldsymbol{s}^2, \boldsymbol{s}^3, \dots, \boldsymbol{s}^{N_m}]$, and aggregate all operation actions in vector $\boldsymbol{O}$. Configuration $\boldsymbol{S}$ can be calculated as $\boldsymbol{S} = \boldsymbol{BO}$, where matrix $\boldsymbol{B}$ records changes to the state of the configuration by operational actions. Operational actions cannot overdraw modules from the current configuration, $\boldsymbol{S_0} - \boldsymbol{B_0 O} \geq \boldsymbol{0}$ (subscript $\boldsymbol{0}$ indicates the current configuration).

The degradation of machine k is characterized by the failure probability of machine k, which is estimated as the maximum module failure probability over all modules forming the machine. The failure probability of machine k is expressed as

$$u^k = \max_i \left(P_{di} s_k^i - P_{pi} p_a^i\right), \tag{1}$$

where P_{di} are module failure probabilities, and P_{pi} are reductions in module failure probabilities by maintenance. The optimization model is formulated:

$$\begin{aligned} \min_{\boldsymbol{o},\boldsymbol{p},\boldsymbol{q}} \ & \textstyle\sum_{k=1}^{k=N_s} w_{dk} u^k + \sum_{i=1}^{i=N_m} \boldsymbol{c}_{oi}^T \boldsymbol{o}^i + \sum_{i=1}^{i=N_m} \boldsymbol{c}_{qi}^T \boldsymbol{q}^i + \sum_{i=1}^{i=N_m} \boldsymbol{c}_{pi}^T \boldsymbol{p}^i \\ \text{s.t.} \ & (a)\ \textstyle\sum_{i=1}^{i=N_s} \boldsymbol{t}_{oi}^T \boldsymbol{o}^i + \sum_{i=1}^{i=N_m} \boldsymbol{t}_{qi}^T \boldsymbol{q}^i + \sum_{i=1}^{i=N_m} \boldsymbol{t}_{pi}^T \boldsymbol{p}^i \leq \bar{t} \\ & (b)\ \boldsymbol{S_0} - \boldsymbol{B_0 O} \geq \boldsymbol{0},\ \boldsymbol{M}_v \boldsymbol{S} = f_{mr}(\boldsymbol{v}, \boldsymbol{n}) \qquad (2) \\ & (c)\ \boldsymbol{o},\ \boldsymbol{q},\ \boldsymbol{p} \text{ are binary variables,} \end{aligned}$$

where $\bar{t}$ is the available time, w_{dk} is the importance of machine k (which is related to the number of machines at the same stage as machine k), $\boldsymbol{c}_x$ denotes the increase in module failure probability due to action x, and $\boldsymbol{t}_x$ denotes the time for action x. $\boldsymbol{M}_v$ is a binary matrix that defines the modules in each machine. f_{mr} calculates the modules required for all machines in a configuration given by machine type $\boldsymbol{v}$ and number $\boldsymbol{n}$ of machines at each stage. Constraint (a) ensures that the ADLM actions are done in the available time; (b) ensures that the scheduled actions can be done with the available resources. The model represents the three main operations of interest in the reconfigurable manufacturing system: maintenance ($\boldsymbol{v} = \boldsymbol{v_0}, \boldsymbol{n} = \boldsymbol{n_0}$), repair ($\boldsymbol{v} = \boldsymbol{v_0}, \boldsymbol{n} > \boldsymbol{n_0}$) and reconfiguration ($\boldsymbol{v} \neq \boldsymbol{v_0}, \boldsymbol{n} \neq \boldsymbol{n_0}$) where $\boldsymbol{v}, \boldsymbol{n}$ are machine type and number at all stages, and subscript $\boldsymbol{0}$ indicates current time. Maintenance is triggered once spare time is available. Reconfiguration and repair decisions are triggered by events such as unfulfilled demands and module failure. Reconfigurations require the new configuration to be determined before planning for operational actions.

2.2. Generation of Candidate Configurations

To generate candidate configurations, the approach of the authors in [14] is modified to generate configurations $(\boldsymbol{v}, \boldsymbol{n})$ with large capacity that have also low module failure probability, i.e.

$$\begin{aligned} \max_{\boldsymbol{v},\boldsymbol{n}} \ & \textstyle\sum_{k=1}^{k=N_p} w_{pk} f_{pr}(k, \boldsymbol{v}, \boldsymbol{n}) - w_t f_{rt}(\boldsymbol{v}, \boldsymbol{n}, \boldsymbol{v_0}, \boldsymbol{n_0}) - w_p f_p(\boldsymbol{v}, \boldsymbol{n}, \boldsymbol{P_d}) \\ \text{s.t.} \ & (a)\ f_{m_p}(\boldsymbol{v}, \boldsymbol{n}) \leq s_{m_p}(t), \quad \forall p \qquad (3) \\ & (b)\ 1 \leq v_{t_s} \leq N_{sv},\ 0 \leq n_{t_s} \leq N_{sn} \text{ and integer}, \ \forall t_s, \end{aligned}$$

where f_{pr} represents an optimization model that calculates the maximum capacity for part k, f_{rt} is an optimization model that evaluates the minimum reconfiguration time ignoring module degradation, f_p is a function that estimates the overall failure probability of a configuration. Weight factors w_x are used to add preferences to the configuration candidates, i.e., higher capacity for some of the parts, quicker reconfiguration or healthier configuration. The approach generates a population of candidate configurations $\boldsymbol{C} = [\boldsymbol{c}^1, \boldsymbol{c}^2, \dots, \boldsymbol{c}^{N_{cd}}] = [(\boldsymbol{v}^1, \boldsymbol{n}^1), \dots, (\boldsymbol{v}^{N_{cd}}, \boldsymbol{n}^{N_{cd}})]$, which are used in making reconfiguration decisions.

2.3. Evolution of Capacity Distribution: Uncertainty Propagation

Consider a configuration c_0 with capacity pr_0. When a module of a machine fails, that machine fails, and configuration c_0 is damaged. All the remaining undamaged machines in configuration c_0 can be thought of as another configuration c_i. Without repair or reconfiguration, configuration c_i continues production until one of this modules fails. This process is repeated until the system completely loses its capacity. A robust configuration is one where the probability of satisfying the demand is high even when some modules fail. Robustness is critical once unexpected damage occurs and demand is urgent.

The number N_d of possible damaged configurations that can result from one single heathy configuration is $2^{N_s} - 1$ which makes exhaustive analysis of all damaged configurations intractable. Thus, the evolution of capacity is used as an alternative approach. Considering that damage can only lower the capacity of a configuration, the capacity evolution over time is

$$\begin{bmatrix} \mathrm{P}(pr_0(t+1)) \\ \mathrm{P}(pr_1(t+1)) \\ \mathrm{P}(pr_2(t+1)) \\ \vdots \\ \mathrm{P}(pr_N(t+1)) \end{bmatrix} = \begin{bmatrix} A_{00} & 0 & 0 & & 0 \\ A_{01} & A_{11} & 0 & \cdots & 0 \\ A_{02} & A_{12} & A_{22} & & 0 \\ & \vdots & & \ddots & \vdots \\ A_{0N} & A_{1N} & A_{2N} & \cdots & 1 \end{bmatrix} \begin{bmatrix} \mathrm{P}(pr_0(t)) \\ \mathrm{P}(pr_1(t)) \\ \mathrm{P}(pr_2(t)) \\ \vdots \\ \mathrm{P}(pr_N(t)) \end{bmatrix}, \quad (4)$$

where $P(x)$ represents the probability that the RMS capacity is x, $pr_i(t)$ decreases as i increases, N represents the number of different capacities of all configurations, and $pr_N(t)$ =0. It is assumed that module failures are independent stochastic variables. Coefficients A_{ij} are obtained as follows:

1: Compute original and damaged configuration capacities by f_{pr}.
2: Cluster all configurations by capacity.
3: For each configuration c_i (with n_{t_s} machines at stage t_s and non-zero capacity), compute the probability $P_{c_i c_j}$ that one of its modules will fail and become configuration c_j (with n_{m,t_s} machines at stage t_s and non-zeros capacity) by using

$$P_{c_i c_j} = \prod_{t_s=1}^{t_s=N_s} \sum_k^{n_{t_s}} (1-p_{dk})^{(1-I_{dk})} (p_{dk})^{(I_{dk})}, \quad (5)$$

where I_{dk} is a binary value that records if machine k is damaged (1) or not (0), and hence $\sum_{k=1}^{k=n_{t_s}} I_{dk} = n_{m,t_s}$, and p_{dk} is the failure probability for machine k, calculated as

$$p_{dk} = 1 - \prod_{i=1}^{i=m_{v_k}} (1 - p_{m_i}), \quad (6)$$

where m_{v_k} is the number of modules in machine k, and p_{m_i} is the failure probability of module i.

4. Compute coefficients A_{ij} as

$$A_{ij} = p_{pr_i pr_j} = \sum_i^{N_c} \sum_{j \neq i}^{N_c} I_{c_i c_j} p_{c_i c_j}, \forall i \in [1, N], \forall j, \quad (7)$$

$$A_{0j} = p_{pr_0 pr_j} = 1 - \sum_{j=1}^{j=N-1} p_{pr_0 pr_j}, \forall j, \quad (8)$$

where $I_{c_i c_j} = 1$ if $pr_i = f_{pr}(\boldsymbol{c_i})$ & $pr_j = f_{pr}(\boldsymbol{c_j})$, and 0 otherwise.

Capacities pr_i can be grouped in a vector $\boldsymbol{pr}$. Starting from a configuration with capacity pr_0, the probabilities of available capacity to be the entries in $\boldsymbol{pr}$ over a time horizon $t = 0, \ldots, h$ are

$$\mathrm{P}(\boldsymbol{pr}(t+h)) = \boldsymbol{A}^h \, \mathrm{P}(\boldsymbol{pr}(t)) = \boldsymbol{A}^h [1\ 0\ 0 \ldots 0]^T. \quad (9)$$

2.4. Configuration Planning

Given the evolution of capacity probability over time, the performance of a candidate configuration is measured by the probability of satisfying the demand received in the planning horizon. First, the processing time $pt_k(t)$ for each part k is calculated. The number of parts k processed between time t and $t+1$ is $\phi_k(t) = pr_k(t) pt_k(t)$. The variance of $\phi_k(t)$ is given by

$$\mathrm{Var}[\phi_k(t)] = \mathrm{Var}[pr_k(t) pt_k(t)] = pt_k^2(t) \mathrm{Var}[pr_k(t)], \quad (10)$$

$$\mathrm{Var}\left[\sum_{\tau=t+1}^{\tau=t+t_p} pr_k(\tau) pt_k(\tau)\right] = \boldsymbol{pt}_k^T \boldsymbol{\Sigma^k} \boldsymbol{pt}_k, \quad (11)$$

where $\boldsymbol{\Sigma^k}$ is the capacity covariance matrix with entries given by $\Sigma_{ij}^k = cov(pr_k(\tau), pr_k(\tau')) = \mathrm{E}(pr_k(\tau)\, pr_k(\tau')) - \overline{pr_k}(\tau)\, \overline{pr_k}(\tau')$.

The processing time can be computed by minimizing $\mathrm{Var}[\phi_k(t)]$:

$$\min_{\boldsymbol{pt}} \quad \sum_k \boldsymbol{pt}^T \boldsymbol{\Sigma}^k \boldsymbol{pt} \quad (12)$$

$$\text{s.t.} \quad (a) \sum_{\tau=t+1}^{t_{n_d}} \overline{pr_k}(\tau) pt_k(\tau) \geq \sum_{\tau=t+1}^{t_{n_d}} d_k(\tau), \ \forall n_d, \forall k$$

$$(b) \ [\textstyle\sum_k pt_k(\tau) + rt(\tau)] \leq \bar{t} - rt(\tau), \ pt_k(\tau) \geq 0 \ \ \forall \tau,$$

where constraint (a) ensures that all demands received can be satisfied by the average capacity, and constraint (b) ensures that the total processing is done in the available time. As uncertainty increases with time, the model prefers to process the parts early in the planning horizon. The vector of processing times is updated at each time step according to the new demand received and according to the current system status.

The configuration with the lowest failure probability is selected, by solving the following binary integer-programming problem

$$\min_z \quad \sum_k \sum_{n_d} \alpha_{k,n_d}(\boldsymbol{c}^z), \quad (13)$$

where $\boldsymbol{c}^z \in \boldsymbol{C}$, and α_{k,n_d} describes the marginal failure probability for satisfying demand n_d for part k, which follows conditions

$$\mathrm{P}\left(\sum_{\tau=t+1}^{t_{n_d}} pr_k(\tau) pt_k(\tau) \geq \sum_{\tau=t+1}^{t_{n_d}} d_k(\tau)\right) \geq 1 - \alpha_{k,n_d}. \quad (14)$$

The total demand before the due time is $\lambda_{k,n_d} = \sum_{\tau=t+1}^{t_{n_d}} d_k(\tau)$, and the total production before the due time is $X_{k,n_d} = \sum_{\tau=t+1}^{t_{n_d}} pr_k(\tau) pt_k(\tau)$. Because X_{k,n_d} follows a non-Gaussian distribution, Cantelli's inequality and Eq. (14) are used to estimate α_{k,n_d}, which is a metric for configuration performance:

$$\mathrm{P}(X - \mathrm{E}[X] \geq \lambda) \geq \frac{\sigma^2}{\sigma^2 + \lambda^2}, \text{ and thus } \alpha_{k,n_d} = \frac{{\lambda_{k,n_d}}^2}{\sigma_X^2 + \lambda_{k,n_d}^2}. \quad (15)$$

By using this model, the selected configuration will not only satisfy the demand, but will also very likely satisfy the demand even when some of its modules fail. This robustness is important for the RMS operation as the occurrence of damage is usually unpredictable and performing reconfigurations may be ineffective due to time pressure. Following this model, reconfiguration decisions are triggered once one of the following conditions holds: 1) the current configuration cannot satisfy the new demand; 2) the module stocks are changed.

2.5. Module Failure Forecasting

An accurate estimation of failure probability is essential for describing the ability of the module to function under given conditions for a specified period of time. The diversity in reconfiguration actions raises the difficulty of identifying the causes of module failures. The sequence of operational actions also affects the chances of failure, i.e., accumulation of failure probability depends on the sequence of usage the module has experienced. To address this, a data-driven approach is used to estimate the module failure probability by using a deep-learning model with long short-term memory (LSTM). LSTM is superior to other models by its strengths of capturing the dynamic behavior for a time sequence. The model created in this study contains a sequential input layer, an LSTM layer, a hidden layer (with 25 nodes) and a regression layer. For training, all operational actions exerted are recorded as a 'resumé' of the module, including the action type and time. Actions are discretized and recorded as a matrix of binary vectors. The output of the training data is the estimated usage of each module, which is assumed to be measurable. With continuous retraining by the data from daily operation, the model can be calibrated and used for forecasting the damage probability given the resumé of each module.

3. Prototype implementation

The example from previous work of the authors [14] is used to test a prototype implementation of the model. A company invests in a single processing line with a limited number of modules. Two types of parts with multiple features are to be processed. 5 types of machines with multiple variants are available for the processing service. The modules stochastically get damaged based on their usage history. Once damaged, module refills are ordered immediately and arrive in 14 days. The assembly time is 0.2 hours for machine tools and 1 hour for bases or arms. Disassembly times are half of assembly times. The time required for actions is: relocation among stages = 0.2 hours; relocation between a stage and storage = 0.5 hours; maintenance = 4 hours. The time required for ADLM actions is the sum of action times.

4. Results

A parametric study is performed to illustrate the dependence of selected actions and module degradation based on the available time. Figure 2 shows the results of this study.

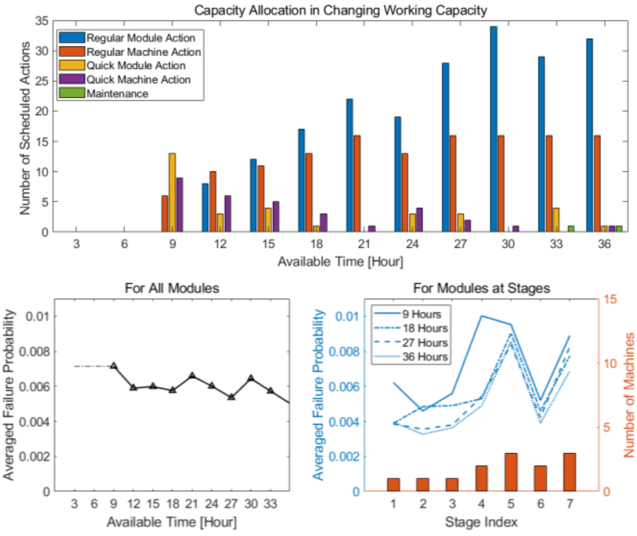

Figure 2. System behaviors change with available working capacity.

To achieve reconfiguration, at least, 6 hours of available time is required. With increasing available time, the decision-making process prefers to replace quick actions with regular ones to protect modules. Once the available time exceeds 33 hours, maintenance is scheduled to improve the condition of modules in storage. A comparison of module degradation across different stages reveals that although a new configuration can be achieved within 6 hours of available time, the new configuration cannot sustain operation with a high capacity. In single-machine stages, high module failure probabilities increase the risk of reduction in RMS capacity. Given sufficient time, the module condition at stages with a few machines are improved resulting in better than average condition for all modules.

The model is then used to simulate the company operating for 100 days. Demand can be sensed about 14 days ahead of due time. Given 48 hours of working capacity, all received demands are satisfied on time by the proposed model. Figure 3 illustrates module conditions, configuration capacity and module stocks during simulation. To reduce the uncertainty in satisfying the demand, it can be observed that the model schedules processing jobs as soon as demands are received. Consequently, module failure rates increase when demands are received. Starting with a configuration with maximized capacity and allowing no reconfiguration, the modules degrade quickly until the configuration completely loses its capacity at Day 62 and requires a very long time to recover with maintenance actions. This stops the RMS operation and incurs a significant amount of missing demands. When reconfiguration is allowed, the system initiates reconfigurations twice, on Day 41 and on Day 71.

Both reconfigurations improve the RMS performance by simplifying the configuration. In each instance, the new configuration requires fewer modules but works at a slightly reduced capacity. Despite these capacity reductions, the RMS keeps up with the demands and also the failure rates are reduced. Although the reduction in capacity requires the system to spend more time processing jobs, which increases failure probabilities slightly, the simplification in configuration makes the maintenance easier and efficient. As a result, module failure probabilities decrease gradually once demand is fulfilled. The reduction in module number makes the configuration robust, not only for reducing the usage of modules, but also, for an increased availability of module stock for maintenance and replacement.

Changes of ADLM and reorder times may impact the RMS unavailability period due to the system reconfiguration, repair and maintenance, which may change the quantitative results.

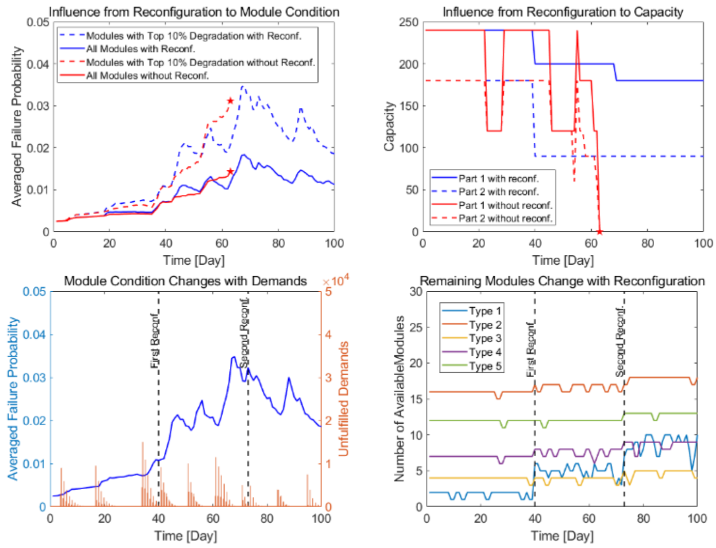

Figure 3. System behaviors change with predictive reconfiguration.

5. Conclusions

The novel model proposed in this paper shows that by taking degradation of machine modules into account, it is possible to create robust reconfigurable manufacturing systems that can cope with damaged modules and deliver demands on time with good maintainability. The approach combines an AI predictive model of degradation with optimization models of maintenance and reconfiguration to achieve these results. The current model initiates reconfiguration reactively and in response to new demand or changes in machine availability. In future work, the authors will explore more proactive reconfiguration triggers as well as deployment in real production environments.

References

[1] Koren, Y., Heisel, U., Jovane, F., Moriwaki, T., Pritschow, G., Ulsoy, G., Van Brussel, H., 1999, Reconfigurable manufacturing systems, CIRP Annals, 48/2:527-540.
[2] Koren, Y., Shpitalni, M., 2010, Design of reconfigurable manufacturing systems, Journal of Manufacturing Systems, 29/4:130-141.
[3] Huang, S., Wang, G., Shang, X. and Yan, Y., 2018, Reconfiguration point decision method based on dynamic complexity for reconfigurable manufacturing systems, Journal of Intelligent Manufacturing, 29/5:1031-1043.
[4] Prasad, D., Jayswal, S., 2019, A review on flexibility and reconfigurability in manufacturing systems, Innovations in Materials Science and Engineering,187-200. Springer, Singapore.
[5] Koren, Y., Gu, X., Guo, W., 2018, Reconfigurable manufacturing systems, Frontiers of Mechanical Engineering, 13/2:121-136.
[6] Terkaj, W., Tolio, T., Urgo, M., 2015, A virtual factory approach for in situ simulation to support production and maintenance planning, CIRP Annals, 64/1:451-454.
[7] Matyas, K., Nemeth, T., Kovacs, K., Glawar, R., 2017, A procedural approach for realizing prescriptive maintenance planning in manufacturing industries, CIRP Annals, 66/1:461-464.
[8] Vogl, G.W., Donmez, M.A., 2015, A defect-driven diagnostic method for machine tool spindles, CIRP Annals, 64/1:377-380.
[9] Iijima, H., Takata, S., 2016, Condition based renewal and maintenance integrated planning. CIRP Annals, 65/1:37-40.
[10] Matyas, K., Nemeth, T., Kovacs, K., Glawar, R., 2017, A procedural approach for realizing prescriptive maintenance planning in manufacturing industries. CIRP Annals, 66/1:461-464.
[11] Gao, R., Wang, L., Teti, R., Dornfeld, D., Kumara, S., Mori, M., Helu, M., 2015, Cloud-enabled prognosis for manufacturing. CIRP Annals, 64/2:749-772.
[12] Cho, S., May, G., Tourkogiorgis, I., Perez, R., Lazaro, O., Maza, B., Kiritsis, D., 2018, A hybrid machine learning approach for predictive maintenance in smart factories, *Conference on Advances in Production Management Systems*:311-317.
[13] El-Maraghy, W.H., Meselhy, K.T., 2009, Quality and maintainability frameworks for changeable and reconfigurable manufacturing, Changeable and Reconfigurable Manufacturing Systems:321-336.
[14] Li, X., Bayrak, A. E., Epureanu, B. I., Koren, Y., 2018, Real-time teaming of multiple reconfigurable manufacturing systems, CIRP Annals, 67/1:437-440.